ProDAQ

User Manual

ProDAQ 3806
Counter/Timer/Frequency Meter
Function Card

PUBLICATION NUMBER: 3806-XX-UM-0100

Copyright, © 2002, Bustec Production, Ltd.

Bustec Production, Ltd.
World Aviation Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

bustec

PROPRIETARY NOTICE

This document and the technical data herein disclosed, are proprietary to Bustec Production Ltd., and shall not, without express written permission of Bustec Production Ltd, be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than Bustec Production Ltd. The information herein has been developed at private expense, and may only be used for operation and maintenance reference purposes or for purposes of engineering evaluation and incorporation into technical specifications and other documents, which specify procurement of products from Bustec Production Ltd.. This document is subject to change without further notification. Bustec Production Ltd. Reserve the right to change both the hardware and software described herein.

Table of Contents

1. INTRODUCTION 5

2. INSTALLATION 7

2.1 Installing a ProDAQ Function Card 8
2.2 Removing a ProDAQ Function Card 10

3. THEORY OF OPERATION 11

3.1 General description 11
3.2 Pulse counters 14
3.3 Time interval counters 15
3.3.1 Example 1 17
3.3.2 Example 2 18
3.3.3 Example 3 18
3.3.4 Example 4 19
3.3.5 Example 5 20
3.4 Hardware configuration 21
3.4.1 Front-end configuration 21
3.4.2 GATE selection 21
3.4.3 Input trigger selection 21
3.4.4 Output trigger configuration 22

4. INPUT AND OUTPUT SIGNALS 23

4.1 Lemo connectors configuration 23
4.2 SCSI connector pin assignment 23

5. TECHNICAL SPECIFICATION 24

6. REGISTER DESCRIPTION 26

6.1 Address Map and Registers 26
6.2 Register Description 27
6.2.1 FCID_REG 27
6.2.2 FCVER_REG 27
6.2.3 FCCTRL_REG 27
6.2.4 FIFOCTRL_REG 31
6.2.5 COMMAND_REG 32
6.2.6 OTRI_REG 32
6.2.7 ITRI_REG 36
6.2.8 DAC_REG 37
6.2.9 MODE_REG 38
6.2.10 IGATEx_REG 39
6.2.11 CHNx_CFG_REG 40
6.2.12 CHN1_2ECNT_REG 43
6.2.13 CHN3_4ECNT_REG 43
6.2.14 CHN5_6ECNT_REG 44
6.2.15 CHNx_PCNT_REG 45
6.2.16 FECONF_REG 46
6.2.17 FIFO_REG 46

Copyright, © 2002 Bustec Production Ltd.

Table of Figures

Figure 1 - Removing the ProDAQ Module Cover 7
Figure 2 - The module assembly..... 9
Figure 3 - Block Diagram 12
Figure 4 - ProDAQ 3806 Main States..... 13
Figure 5 - Operation of the Pulse Counters..... 14
Figure 6: TICNT's modes of operation - example 1..... 17
Figure 7: TICNT's modes of operation - example 2..... 18
Figure 8: TICNT's modes of operation - example 3..... 18
Figure 9: TICNT's modes of operation - example 4..... 19
Figure 10: TICNT's modes of operation - example 5..... 20
Figure 11: Front-End configuration..... 21
Figure 12: GATE selection scheme..... 21
Figure 13: Input trigger selection..... 21
Figure 14: Output trigger configuration..... 22

 Copyright, © 2002 Bustec Production Ltd.

1. Introduction

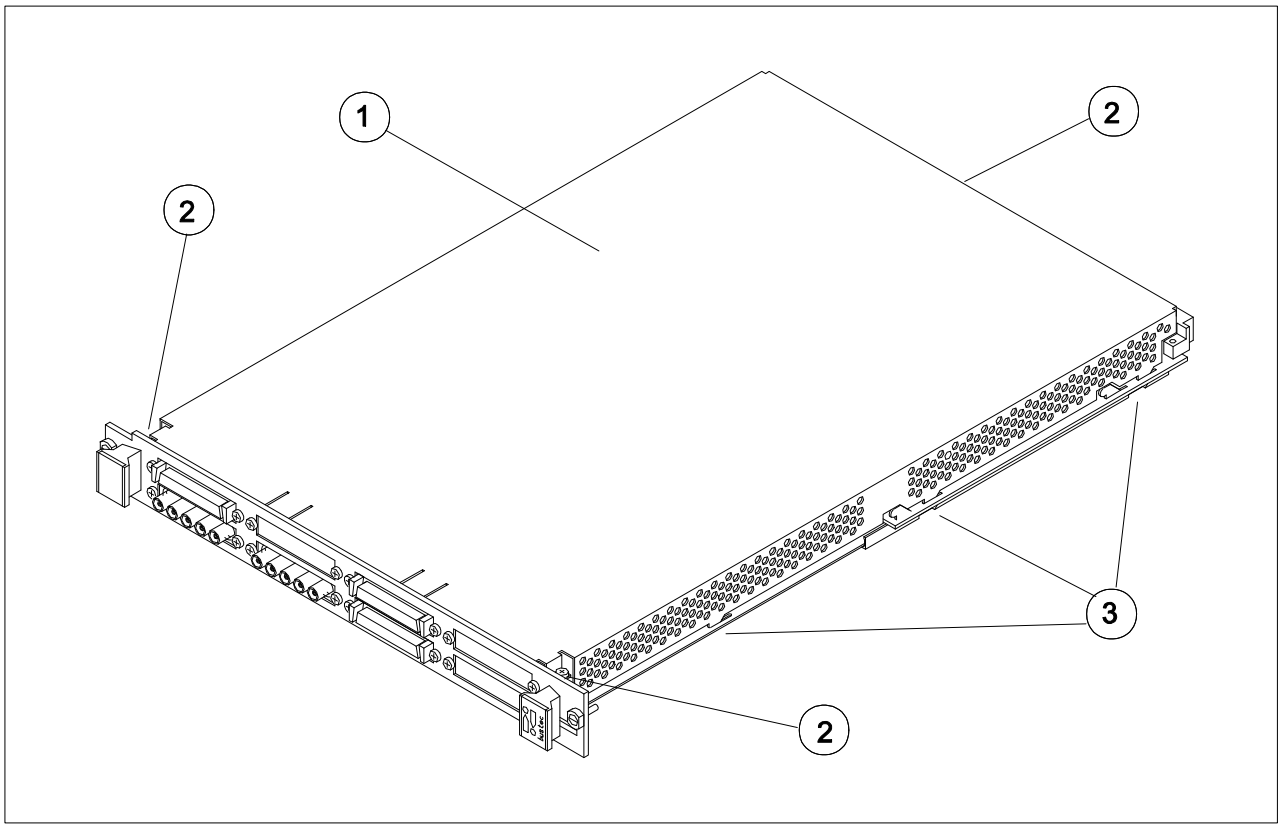
The ProDAQ 3806 Counter Timer Function Card is one of the ProDAQ high-density cards, which can be fitted into ProDAQ Motherboards.
The ProDAQ 3806 Counter Timer Function Card is designed to operate in one of three modes: pulse counter, frequency meter or time interval counter.
In the Pulse Counter Mode and Frequency Measurement Mode incoming signals of up to 25 MHz can be processed on each of the six channels. A time interval counter range is divided into six sub-ranges covering times from 20ns to 10000s with a maximum resolution of 12.5ns. For pulse width measurements the same ranges apply.
Every channel of the 3806 Counter Timer Function Card has two different counters: a 32-bit pulse counter and a 24-bit time interval counter allowing simultaneous operation as a pulse and a time interval counter.
The time interval counter can count consecutive time intervals and store the results in a FIFO memory. This allows the time interval averaging to be performed resulting in increased time measurement accuracy. Up to 128 time values per channel can be stored in the onboard FIFO. ”On-the-fly” readouts can be performed during the measurement cycle.
The time interval can be defined by selecting either rising, falling or consecutive edges of the input signal.

Copyright, © 2002 Bustec Production Ltd.

Copyright, © 2002 Bustec Production Ltd.

2. Installation

To install a ProDAQ Function Card in a ProDAQ Motherboard, you need to remove the modules top cover.



1 - Module Cover 2 - Cover Screws 3 - Cover Hooks

Figure 1 - Removing the ProDAQ Module Cover

To remove the top cover, remove the undercut flathead and two panhead screws that hold the cover in place and remove the cover by sliding it out of its position towards the VXIbus connectors and up. Take special care about the hooks holding it into place. Try not to lift the cover straight up. See figure 1 for the location of the screws.

To re-install the cover, slide it back into its position by placing the small hooks over their holes and moving the cover down and forward. Secure the top cover using the three undercut flathead screws.

Copyright, © 2002 Bustec Production Ltd.

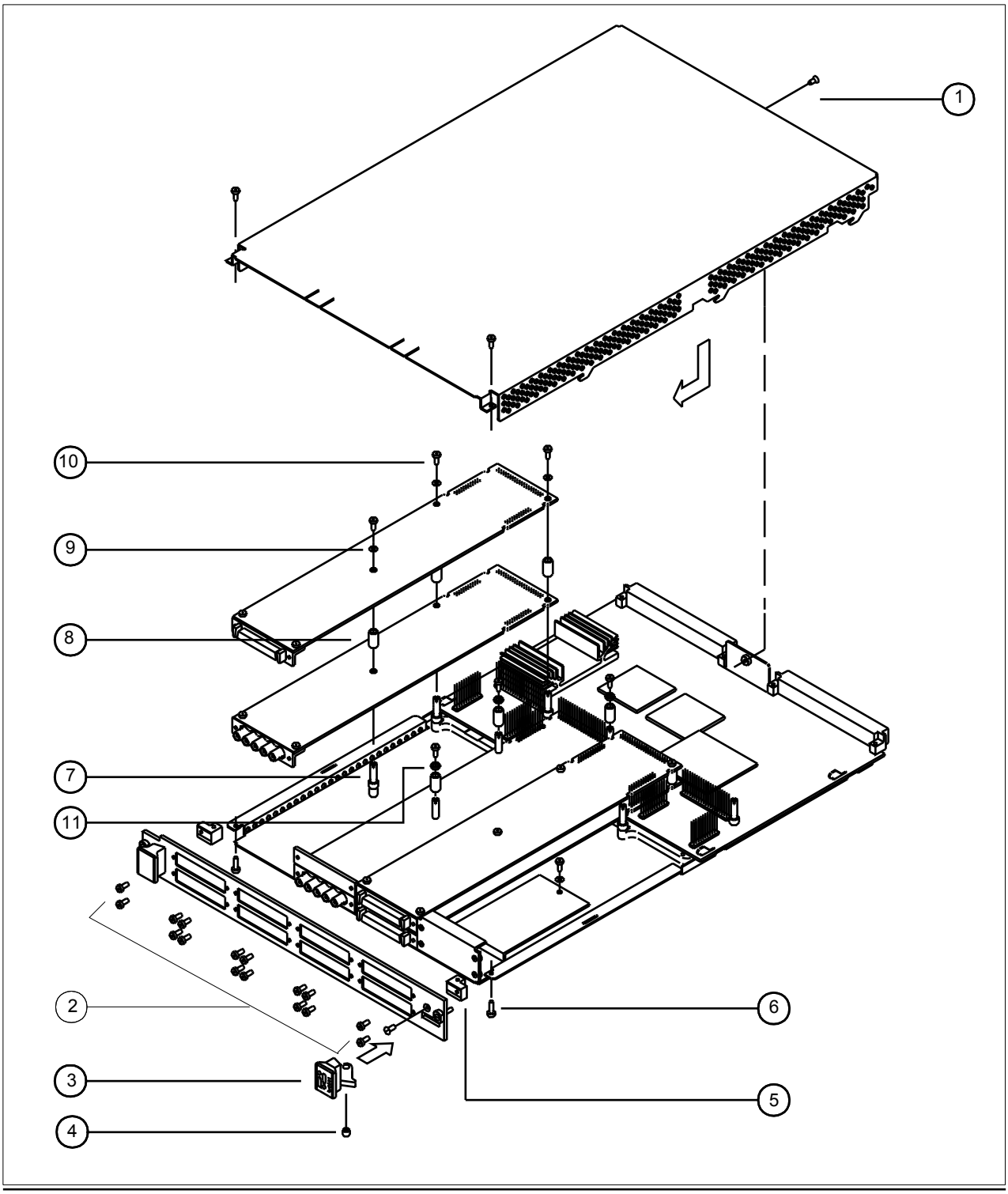
2.1 Installing a ProDAQ Function Card

The ProDAQ Function Cards are arranged inside the ProDAQ Module in four stacks of two cards each. The function cards are mounted face down, e.g. the front-panel connectors as well as the motherboard connectors are underneath the PCB.

To install a ProDAQ Function Card in any of the possible positions, use the following procedure (See figure 3 for reference):

- Remove the top cover of the module as described earlier in this chapter (Fig. 2, Pos. 1).
- Remove all screws on the front-panel holding installed function cards or double filler panels in place (Fig. 2, Pos. 2). Screws holding single filler panels don't need to be removed.
- Remove the two panhead screws that mount the front panel to the modules bottom cover (Fig. 2, Pos. 6).
- Please take special care of the module handles and the rings (Fig. 2, Pos. 3 and 4), which are also fixed by those screws. The mounting angle (Fig. 2, Pos. 5) stays fixed to the front panel.
- Remove the front panel by moving it forward carefully so as to avoid bending the installed function cards.
- Choose the stack and position (lower or upper) where you want to mount the function card. If the stack, in which the function card should be installed, is covered by a double filler panel, you have to remove it before installing the function card.
- Remove the three 2.5mm panhead screws and the crinkle washers from the stack's standoffs (Fig. 2, Pos. 9 and 10 for example).
- If you want to install a function card in the upper position of a stack without having a function card in the lower position, you need to mount both spacers (Fig. 2, Pos. 11) on each standoff. If the stack is already populated with a function card in the lower position, mount only the bigger spacer (Fig. 2, Pos. 8) onto each standoff.
- Place a bayonet (supplied) on each standoff. Align the function card over these and slide carefully down. The function card should be held parallel to the modules bottom cover all the time during its way down.
- Fix the function card by mounting the three 2.5mm panhead screws and the crinkle washers onto each standoff. If you install a function card in the lower position of a stack, you need first to mount both spacers (Fig. 2, Pos. 11) onto each standoff.
- Re-mount the modules front-panel. If there is only one function card mounted in a stack, cover the remaining opening in the front panel by a single filler panel.
- Re-mount the modules top cover.

 Copyright, © 2002 Bustec Production Ltd.



1 - 2.5mm Panhead Screws
2 - 2.5mm Panhead Screws
3 - Module Handle
4 - Ring
5 - Mounting Angle
6 - 2.5mm Panhead Screws
7 - Standoff
8 - Spacer
9 - Crinkle Washer
10 - 2.5mm Panhead Screw
11 - 2mm Spacer

Figure 2 - The module assembly

Copyright, © 2002 Bustec Production Ltd.

2.2 Removing a ProDAQ Function Card

Removing a ProDAQ Function Card is exactly the reverse operation then installing it. After removing the top cover and the front panel as described previously, remove the three roundhead screws that fix the function card(s) on the standoffs.
Take special care when removing the function card(s) not to bend the motherboard connectors.

After removing the function card(s), install the correct combination of spacers on the standoffs. If a stack is populated with only one function card, each of the standoffs needs to be mounted with both spacers to cover the distance between the cards as well as the PCB thickness of the missing card. If a stack is populated with two function cards, only the bigger spacer must be mounted.

Fix any remaining function card again by mounting the three panhead screws on the standoffs, re-mount the front panel and the modules cover.

Copyright, © 2002 Bustec Production Ltd.

3. Theory of Operation

3.1 General description

The ProDAQ 3806 6-channel Counter/Timer/Frequency Meter Function Card provides the following functions:

- Pulse counting – every channel counts incoming pulses within a specified time defined by a gate signal. The gate signal is common to all channels and can be either external or internal. The width of an internally generated gate signal is software selectable from 400ns to 1717s.
- Frequency measurement – as a result of the pulse counting for known gate width.
- Time interval measurement – every channel is capable of measuring time intervals between rising edges, falling edges or consecutive edges of an input signal. If the FIFO is emptying at the end of measurement, up to 256 samples per channel can be stored in an on-board memory. A “read on-the-fly” mode allows data transfer to the host during the measurement thus yielding a substantial increase of the number of time tags per channel. The time interval measurements can be initiated by a trigger signal. The ProDAQ 3806 can accept either one common or 6 independent external trigger signals.
- Pulse width, period and duty cycle measurements – as a result of the time interval measurement capabilities.
- Time interval, pulse width, period and duty cycle averaging.

Every channel of the ProDAQ 3806 Function Card has two different counters: 32-bit pulse counter and 24-bit time interval counter, allowing simultaneous operation as a pulse and a time interval counter.
In the Pulse Counter incoming signals of up to 25 MHz can be processed on each of the six channels. A time interval counter range is divided into six sub-ranges covering times from 40ns to 20000s with a maximum resolution of 12.5ns.
The pulse and time interval counters are globally enabled by the gate signal, which can be generated either internally, externally or by software. In addition, the trigger signal (common or dedicated to every channel) can be used to gate the time interval measurements. Thus provide two modes of operation: a window mode, in which the trigger enables measurements for the time duration of the trigger pulse and a launch mode, in which the trigger starts the measurements. Each mode can be programmed independently on a channel-to-channel basis.
Input signals can be either DC or AC coupled with 1MΩ or 50Ω termination. Every channel has a dedicated 10-bit digital-to-analogue converter to set a threshold within the full-scale input signal range of ±5V.

Copyright, © 2002 Bustec Production Ltd.

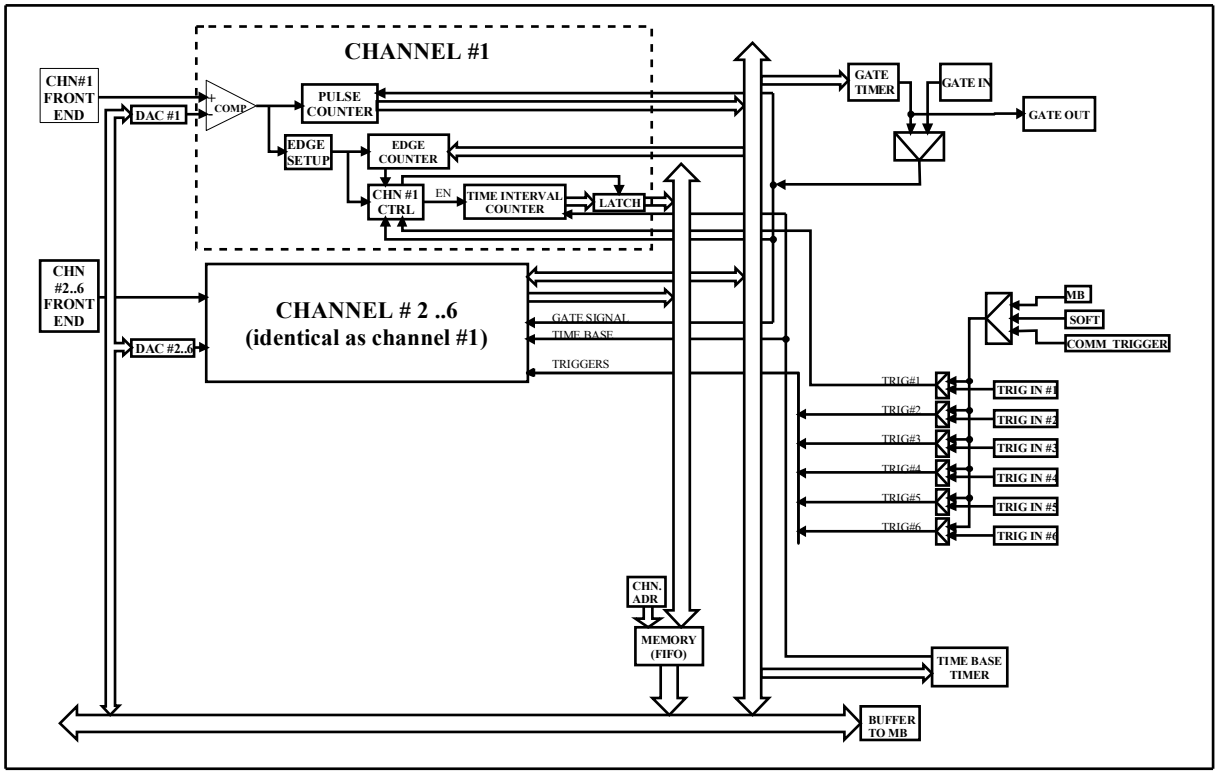


Figure 3 - Block Diagram

The ProDAQ 3806 Counter Timer Function Card houses the following blocks:

PULSE COUNTER counts input signal pulses if the gate is on. The pulse counter can be programmed to count rising edges or falling edges. The full revolution of the counter is treated as an error and the information about that is latched in appropriate register. The pulse counter is a 32-bit wide.

TIME INTERVAL COUNTER counts pulses coming from the Time Base generator (a programmable reference clock). When started the time interval counter counts these pulses continuously and stores the current value of the counter into FIFO when an event happens. An event is an edge of input signal and is software selectable between rising, falling or consecutive edges. The latched data are then stored in a memory.
The time interval counter can be started either by the gate signal or the trigger. In the triggered mode the actual start of the counter can be synchronised to the edge of the input signal. The time interval counter continues its operation until one of the following is reached:

- a pre-set number of measurements has been taken if working in limited mode,
- the end of the trigger window signal if appropriate mode was set,
- the end of the gate signal.

EDGE COUNTER counts down the events (edges) of the input signal. This is a loadable counter and can be programmed to react on either rising, falling or every edge of the input signal. Terminal count of this counter indicates that the pre-set number of measurements has been taken and is used to stop the time interval counter (if in limited mode).

 Copyright, © 2002 Bustec Production Ltd.

TIME BASE TIMER generates the clock for the time interval counters. It can be programmed to generate frequencies from 800Hz to 80MHz.

GATE GENERATOR generates the internal gate signal. The width of the gate is in range from 400ns to 1717s. The internal gate becomes active either upon a software command or upon receiving a trigger signal supplied to the external gate input.

MEMORY is used to store the values form time interval counters when the defined edge of the input signal happens. The latched time interval counter's data and the channel's address are merged together and stored in a FIFO.

The ProDAQ 3806 function card is controlled by the finite state machine (FSM). The FSM can be in one of three states:

ACCESS_state	This is the state after reset and in fact it is an idle state. The configuration and set-up of the card should be done in this state. The counting is disabled. The exit from this state is done using the Arming Command (write to COMMAND_REG).
ARMED_state	The card is armed. This means the configuration is finished and the card is waiting for gate signal to start counting. The edge of gate signal is required to go to the COUNTING_state. Return to ACCESS_state can be performed using FSMreset bit.
COUNTING_state	The enabled counters count until Gate signal is on. The counting is allowed only in this state. Return to ACCESS_state takes place when Gate becomes inactive or FSMreset bit is set or ERROR happened and stopping on errors had been previously enabled. When going to ACCESS_state the COUNTING_END bit is set.

The current state of the CTFC state machine can be read from FCCTRL_REG.

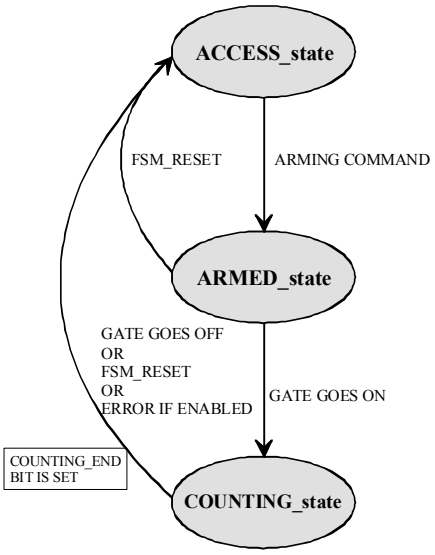


Figure 4 - ProDAQ 3806 Main States

Copyright, © 2002 Bustec Production Ltd.

Before the data acquisition can be started the 3806 has to be set-up properly. Upon a software command the card is armed and ready to acquire a gate signal, which determines the time of measurement.
If the card goes to ARMED_state while the gate is already ON this gate will not be accepted and the card will not go to COUNTING_state until the selected edge of gate happens. It is because the gate circuitry is edge-triggered and that means it requires an edge for its proper operation.
The gate signal enables the pulse counters. The gate signal is necessary for the time interval counters and the TICNTs count only when gate is on but the start of the TICNTs depends on selected mode of operation.
The gate signal source can be selected either as an external, internal generator or software. Internal gate timer can be initiated either by software or an edge of a pulse applied to the Gate Input.
When the gate goes into its inactive state all the counters are disabled. Gate signals will be rejected during ACCESS_state and will not be accepted until the next arming occurs.
Gate signal possible selections are as follow:

GATE			
External	**Internal, programmable pulse**		**Software**
	Software initiated	Triggered from external signal applied to Gate In	

The gate is common for all channels but the modes of operation can be set for every channel independently.

3.2 Pulse counters

In every channel there is a pulse counter. It counts the number of edges of input signal that come when the gate is on. The edge direction that is counted is software selectable to be rising or falling.
The pulse counter is 32-bit wide. It can count the pulses which the frequency is up to 25MHz. Figure 5 shows the explanation of the pulse counter operation. It assumes that the pulse counter was configured to react (to count) to rising edges.

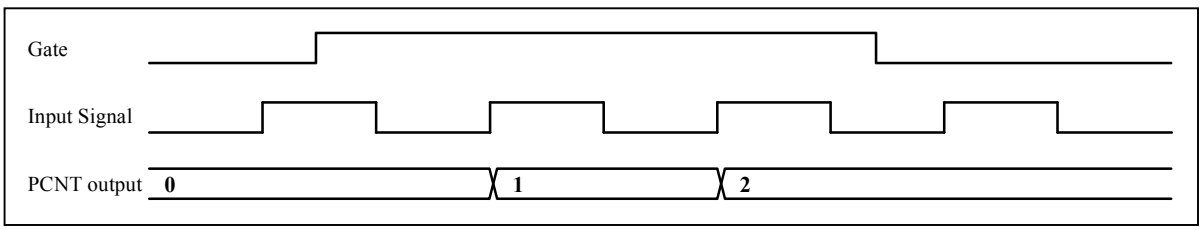


Figure 5 - Operation of the Pulse Counters

The output of the pulse counters is 32-bit wide and has to be read it two steps from PCNTx_REG (were x is the channel number), as a lower and upper 16-bit word. The selection of the word to read is controlled by PCNT_UPWORD bit. If the counter made the full revolution the PCNT error will be reported. The pulse counter can work simultaneously to the time interval counter.

 Copyright, © 2002 Bustec Production Ltd.

3.3 Time interval counters

In every channel there is the time interval counter. When started, the time interval counter counts the number of Time Base pulses. It counts it continuously until it is stopped and therefore many full revolutions can happen. On special moment time called event the momentary counter value is latched and written to the FIFO memory. The event can be defined as a rising edge, falling edge or every edge of the input signal.
There are a few ways of starting the time interval counter and they are presented in the table below.

TICNT STARTED AFTER GATE GOES ACTIVE		**TICNT STARTED AFTER TRIGGER (DURING GATE ACTIVE)**			
Immediately after gate (asynchronously to the input signal)	**After gate on first selected edge of input signal (synchronously to the input signal)**	**Immediately after trigger (asynchronously to the input signal)**		**After trigger on first selected edge of input signal (synchronously to the input signal)**	
		Launch mode	Window mode	Launch mode	Window mode

The TICNT can be configured to start when gate signal goes on. If the asynchronous mode was selected (immediately after starting signal) then the first value latched on the first event will be the interval between the gate edge and the first event. The next values stored in FIFO will be the interval between the events. If the synchronous mode was selected then all values will be the interval between the events.
The TICNT can be configured to start on trigger when gate is on. In this case synchronous/asynchronous mode can be set as well. For the asynchronous mode the first value stored will be the interval between the trigger and first event. In addition, for the trigger the launch or window mode can be selected. The launch mode means that the trigger starts only the counting of the TICNT. The window mode means that as long as trigger is on the counting take place and events are accepted.
There are six time interval counters on the 6-channel board and the samples form the TICNTs are latched on the event and then written to the FIFO. There is only one FIFO memory common for all channels and the samples from all channels are stored there. Of course the arbitration scheme was applied to it. The highest priority is assigned to channel #1, the lowest to channel #6. If there is a request of writing to FIFO set on all channels simultaneously channel #1 will be serviced as the first. It can happen in any channel that waiting for the service, which is for the write to FIFO, new value for this channel is latched on the event overwriting the previous one. It is called as an OVERWRITE error and is indicated on the bit in FCCTRL_REG.
The minimum time interval allowed to measure simultaneously without OVERWRITE error happening depends on the number of channels taking part in data acquisition. The following equation gives the minimum time interval in every channel without OVERWRITE error happening, assuming that the same signal was applied to all of them.

$$MIN\_TIME\_INTERVAL = 2 * Number\_of\_TICNTs * 12.5ns$$

In any case the measured time interval can not be less than 40ns.

Copyright, © 2002 Bustec Production Ltd.

Number of enabled time interval counters	MIN. Time Interval applied to the channels [ns]
1	40
2	50
3	75
4	100
5	125
6	150

The FIFO memory can be read on-the-fly thus allowing unlimited number of samples to be collected. If the FIFO is not emptied on-the-fly the number of samples coming from the channels has to be limited to the FIFO size, that is to 1023 samples in total. Otherwise the OVERWRITE error can happen when FIFO is full (when FIFO is full writing to the FIFO is disabled until there is any place in the FIFO). To limit number of samples the LIMITED mode has to be set and number of samples has to be specified.

Once started the time interval counter will count until one of the following events happens:

STOPPED WHEN:	REMARQUES
Gate becomes inactive	Unconditional ends of counting
Pre-set number of samples was taken	In limited mode only
Trigger becomes inactive	In window mode only, immediately (asynchronously) or synchronised to input signal event

The trigger for the time interval counters can be either common to all channels or independent for each channel. The independent triggers can only be provided by an external source. The common trigger can be one of the following: software, coming from the MB or an external source.

Trigger source for time interval counters:

TRIGGER INDEPENDENT FOR EVERY CHANNEL	TRIGGER COMMON FOR ALL CHANNELS
External	External
	Software
	Coming from the MB

Copyright, © 2002 Bustec Production Ltd.

3.3.1 Example 1

TI counter has been configured to start immediately after gate is on (asynchronous start). The events have been defined as a rising edge. Number of samples was set to 3 in limited mode but for given gate length and input signal timing the same result can be got with unlimited mode.

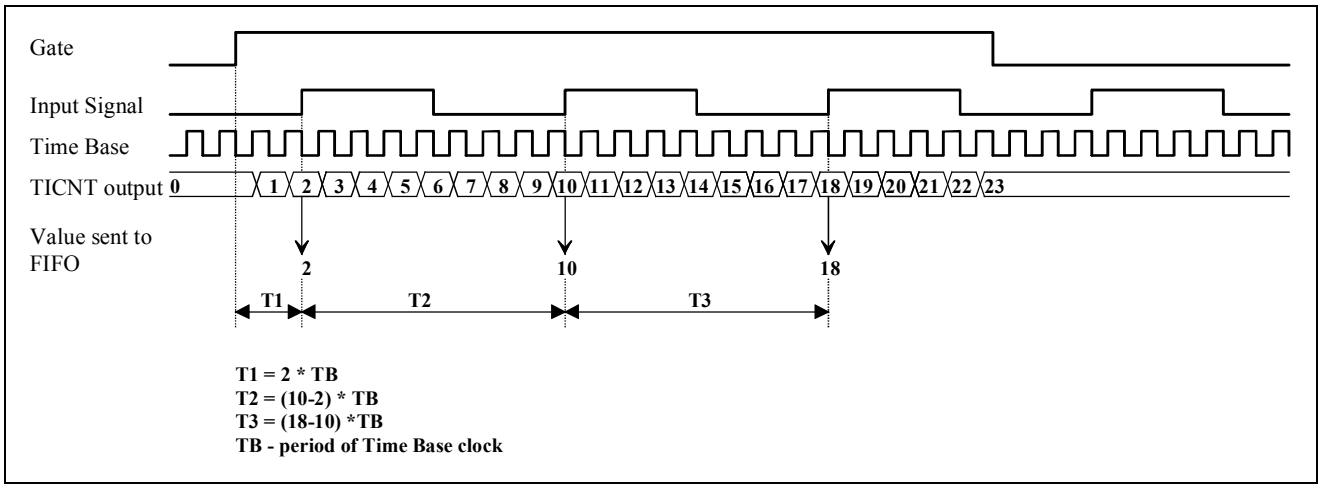


Figure 6: TICNT's modes of operation - example 1

As a result three values (2, 10, 18) will be stored in the FIFO. The first value represents the time interval between Gate edge and first event (rising edge) of input signal:

T1 = 2 * TB.

Every next value has to be processed to get the time interval between two consecutive events (in this case the time interval between two rising edges and in fact period of input signal):

T2 = (10 - 2) * TB = 8 *TB
T3 = (18 – 10) * TB = 8 * TB
where TB is a period of Time Base clock, software selectable.

Copyright, © 2002 Bustec Production Ltd.

3.3.2 Example 2

TI counter has been configured to start after gate is on, synchronously to the input signal event. The events have been defined as a falling edge. Limited mode has been selected and number of samples has been set to 1.

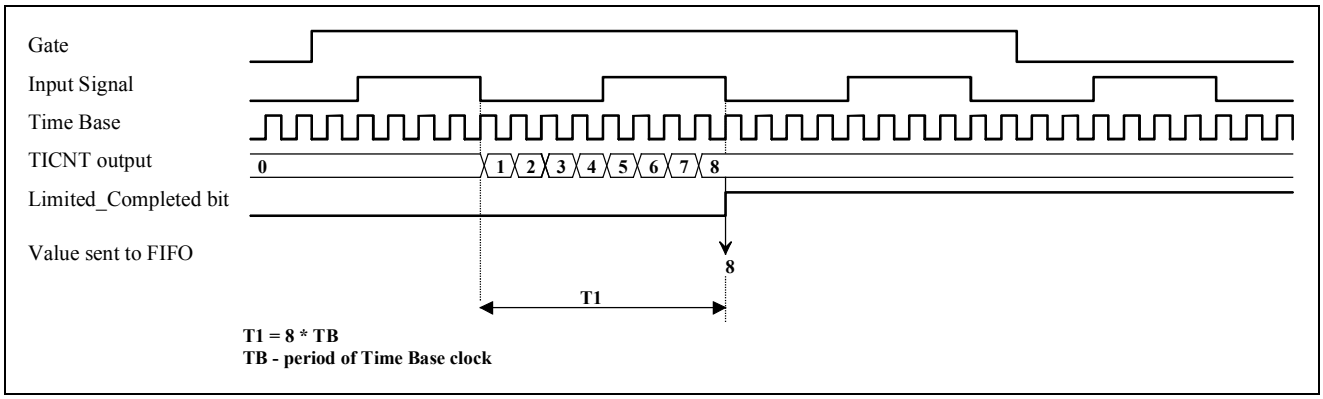


Figure 7: TICNT's modes of operation - example 2

As a result one value (8) will be stored in the FIFO. This value represents the time interval between first event (falling edge) of input signal and second event (falling edge) of the input signal: T1 = 8 * TB.
After the first value is loaded to the FIFO the TI counter is stopped and no more samples will be stored in the FIFO (coming from this TI counter).

3.3.3 Example 3

The time interval counter has been configured to start after trigger (when the gate is on), synchronously to the input signal event. The trigger works in window mode. The events have been defined as an every edge with the falling edge as first. Limited mode has been selected and number of samples has been set to 3 but as shown in
Figure 8 limit wasn't reached.

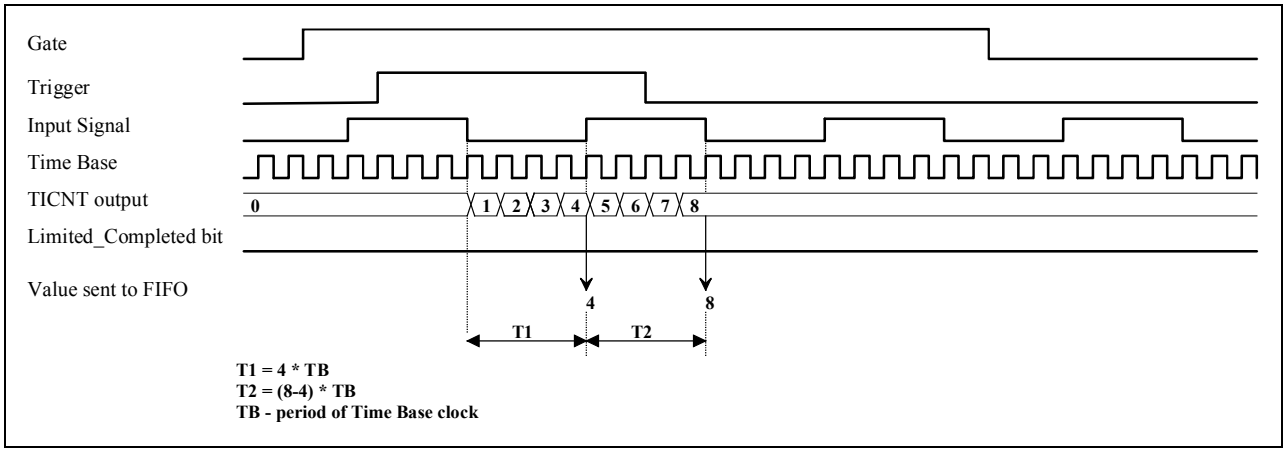


Figure 8: TICNT's modes of operation - example 3

 Copyright, © 2002 Bustec Production Ltd.

As a result two values (4 and 8) will be stored in FIFO. The first value represents the time interval between first event (falling edge) and second event (rising edge): T1 = 4 * TB.
The second value represents the time interval between second event (rising edge) and third event (falling edge): T2 = (8 – 4) * TB.
In fact T1 value is a negative pulse width and T2 is a positive pulse width.
The TICNT was started when gate and trigger were on and was synchronised to input signal edge. The TICNT was stopped when trigger went off and stopping was synchronised to selected input signal edge.
The limit, set to 3, wasn't reached but TI counter was stopped because of the window trigger mode.

3.3.4 Example 4

TI counter has been configured to start immediately after trigger (when the gate is on), asynchronously to the input signal event. The trigger works in launch mode. The events have been defined as an every edge with the falling edge as first. Limited mode has been selected and number of samples has been set to 3.

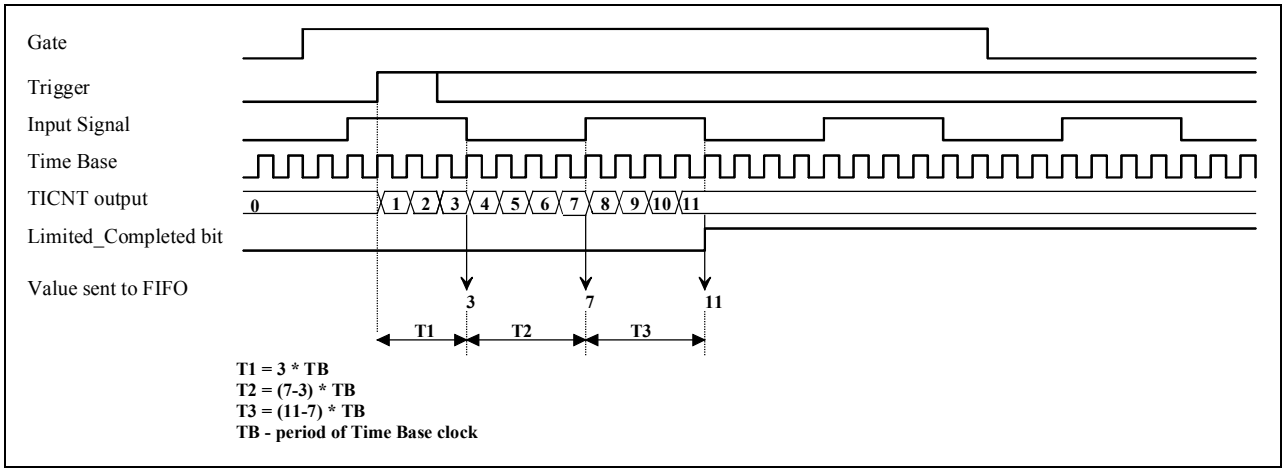


Figure 9: TICNT's modes of operation - example 4

As a result three values will be stored in FIFO: 3, 7 and 11.
The first value represents the time interval between starting edge of trigger and first event (falling edge): T1 = 3 * TB.
The second value represents the time interval between first event (falling edge) and second event (rising edge): T2 = (7 – 3) * TB.
The third value represents the time interval between second event (rising edge) and third event (falling edge): T3 = (11 – 7) * TB.
In fact T1 value is the time interval between trigger and first event, T2 is a negative pulse width and T3 is a positive pulse width.
The TICNT was started when gate and trigger were on, immediately after trigger (asynchronously to input signal edge). The TICNT was stopped after set number of samples was collected because of the limited mode.

Copyright, © 2002 Bustec Production Ltd.

3.3.5 Example 5

TI counter has been configured to start after the gate is on, synchronously to the input signal event.
The events have been defined as a rising edge.
Unlimited mode has been selected.

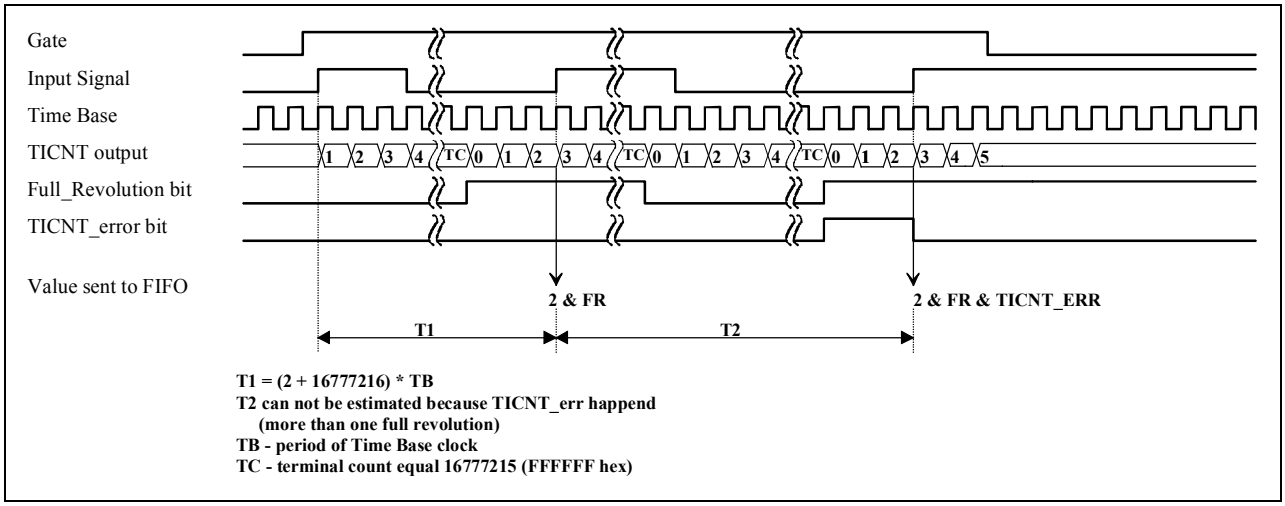

Gate
Input Signal
Time Base
TICNT output
Full_Revolution bit
TICNT_error bit
Value sent to FIFO
1 2 3 4 TC 0 1 2 3 4 TC 0 1 2 3 4 TC 0 1 2 3 4 5
2 & FR
2 & FR & TICNT_ERR
T1
T2
T1 = (2 + 16777216) * TB
T2 can not be estimated because TICNT_err happend
(more than one full revolution)
TB - period of Time Base clock
TC - terminal count equal 16777215 (FFFFFF hex)

Figure 10: TICNT's modes of operation - example 5

As a result two values will be stored in FIFO: (2 & FR) as a first and (2 & FR & TICNT_ERR) as a second.
The first value represents the time interval between the first and second rising edge and because of the full revolution that was made this time interval has to be calculated as follow:
T1 = (2 + 16777216) * TB.
The second value represents the time interval between the second and third rising edge but because of the fact that full revolution happened more than once the TICNT_error bit was set and the T2 can not be estimated (the hardware allows to handle the situation when the full revolution happens no more than once).

 Copyright, © 2002 Bustec Production Ltd.

3.4 Hardware configuration

3.4.1 Front-end configuration

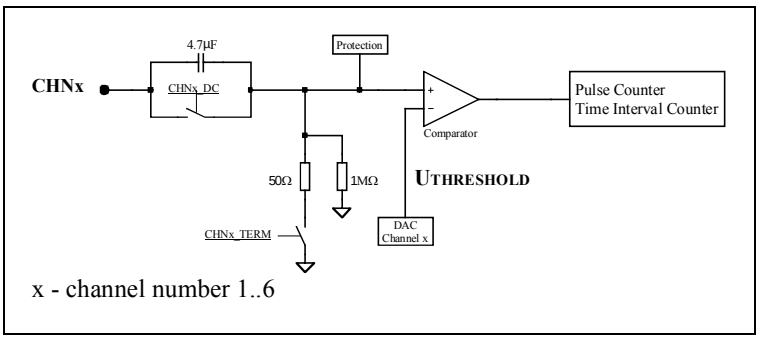


Figure 11: Front-End configuration

3.4.2 GATE selection

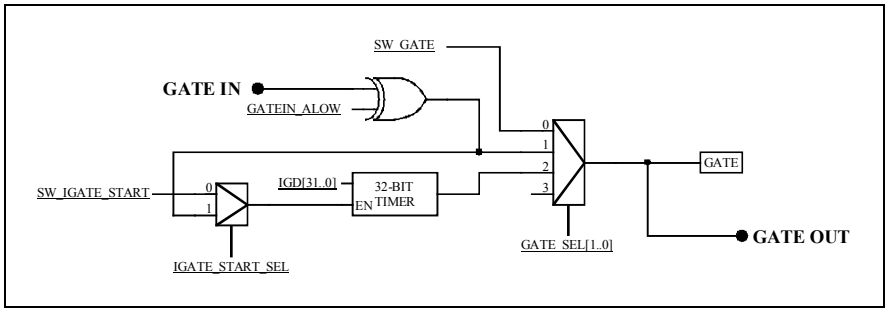


Figure 12: GATE selection scheme

3.4.3 Input trigger selection

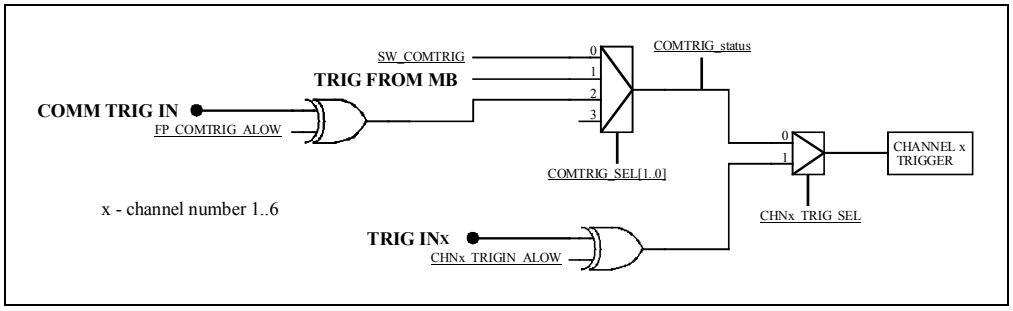


Figure 13: Input trigger selection

Copyright, © 2002 Bustec Production Ltd.

3.4.4 Output trigger configuration

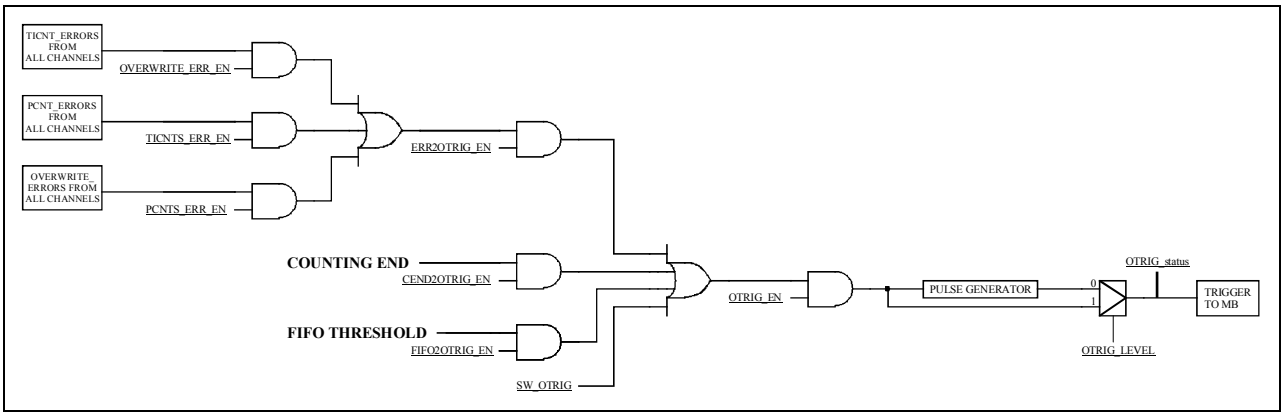


Figure 14: Output trigger configuration

 Copyright, © 2002 Bustec Production Ltd.

4. Input and output signals

4.1 Lemo connectors configuration

#	SIGNAL	DESCRIPTION	LEVEL	I/O	QTY
1	Channel Input (CHN1..6)	Counter / timer inputs	±5V	In	6
2	Gate In (GATE IN)	External gate input or trigger input used to start internal gate	TTL	In	1
3	Common Trigger In (COMM TRIG IN)	External, common for all channels, trigger input. The common trigger in shares the pin with one line of independent trigger in	TTL	In	1
4	Trigger In (TRIG IN 1..6)	External, independent for each channel, trigger inputs. One line of independent trigger in shares the input pin with common trigger in	TTL	In	6
5	Gate Out (GATE OUT)	Signal from internal gate	TTL	Out	1

4.2 SCSI connector pin assignment

Signal	A	B	Signal
GATE OUT	1	26	**GATE IN**
DGND	2	27	DGND
DGND	3	28	**COMM TRIG IN**
DGND	4	29	DGND
DGND	5	30	**Reserved**
DGND	6	31	DGND
Reserved	7	32	**CHN4**
DGND	8	33	DGND
DGND	9	34	**TRIG IN6**
DGND	10	35	DGND
DGND	11	36	**TRIG IN5**
DGND	12	37	DGND
Reserved	13	38	**CHN3**
DGND	14	39	DGND
DGND	15	40	**TRIG IN4**
DGND	16	41	DGND
DGND	17	42	**TRIG IN3**
DGND	18	43	DGND
CHN6	19	44	**CHN2**
DGND	20	45	DGND
DGND	21	46	**TRIG IN2**
DGND	22	47	DGND
DGND	23	48	**TRIG IN1**
DGND	24	49	DGND
CHN5	25	50	**CHN1**

Copyright, © 2002 Bustec Production Ltd.

5. Technical specification

ITEM	SPECIFICATION
Input characteristics	
• Number of Input Channels	6
• Input Range	±5V
• Input Type	Single-ended
• Coupling	DC, AC
• Input Impedance	50Ω or 1MΩ for both DC and AC coupling
• AC Coupling	1µF/25V in series with input signal
• Max. Input Voltage	±10VDC @ 50Ω termination ±20VDC @ 1MΩ termination
Sensitivity	
• Up to 10MHz	25mVrms
• 10MHz to 20MHz	40mVrms
• 20MHz to 25MHz	80mVrms
Frequency Measurement and Pulse Counting	
• Max. frequency	25MHz
• Min. pulse width	20ns
• Counter width	32 bits
Gate Internal	
• Range	400ns to 1717s
• Resolution	400ns
Gate External	
• Level	TTL
• Minimum pulse duration	25ns
• Active edge	Software selectable
Time Interval Measurement	
• Range	40ns to 20000s
• Resolution	12.5ns to 1.25ms in decade steps
• Minimum time interval	40ns
• Counter width	24 bits
Time Base	
• Frequency	80MHz
• Overall stability	±15ppm / 0..50°C
• TCXO option	±1.5ppm
Threshold Level	
• Range	±5V
• Resolution	10 bits
• Accuracy	±50mV

 Copyright, © 2002 Bustec Production Ltd.

Trigger Input	
• Level	TTL
• Min. pulse width	25ns
• Active level	Software selectable
FIFO	1023 samples per 6 channels or unlimited when emptying on-the-fly
Front Panel Connector	50-pin SCSI Female Connector
Current Consumption (typ.)	80 mA @ +12V 50 mA @ -12V 490 mA @ +5V 40 mA @ -5.2V 5 mA @ -2V
Power Consumption (max.)	< 4.4 W
Operating Temperature	0 .. +50°C
Storage Temperature	-40 .. +70°C
Humidity	0-90%, non-condensing
Warm-up time	< 15min.
Dimensions	230mm x 52.6mm
Weight	90 g

Copyright, © 2002 Bustec Production Ltd.

6. Register Description

6.1 Address Map and Registers

All addresses are given in a hexadecimal notation. FC_ADR is address in FC address space. VXI_ADR is address in VXI address space. The appropriate address offset depending on FC position into MB should be applied (refer to MB manual).

FC_ ADR	VXI_ ADR	Register Name		Function
0	0	FCID_REG	RO	FC ID register
1	4	FCVER_REG	RO	FC version register
2	8	FCCTRL_REG	RW C	General control and status register
3	C	FIFOCTRL_REG	RW	FIFO control/status register
4	10	COMMAND_REG	RW	Command register
5	14	OTRI_REG	RW	Output trigger control register
6	18	ITRI_REG	RW	Input trigger control register
7	1C	DAC_REG	RW	DACs setting register
8	20	MODE_REG	RW	Mode register
9	24	IGATEL_REG	RW	Internal gate width register, low
A	28	IGATEH_REG	RW	Internal gate width register, high
B	2C	CHN1_CFG_REG	RW	CHN1 configuration register
C	30	CHN2_CFG_REG	RW	CHN2 configuration register
D	34	CHN3_CFG_REG	RW	CHN3 configuration register
E	38	CHN4_CFG_REG	RW	CHN4 configuration register
F	3C	CHN5_CFG_REG	RW	CHN5 configuration register
10	40	CHN6_CFG_REG	RW	CHN6 configuration register
11	44	reserved		
12	48	reserved		
13	4C	CHN1_2ECNT_REG	RW	CHN1 and CHN2 edge counter setting register
14	50	CHN3_4ECNT_REG	RW	CHN3 and CHN4 edge counter setting register
15	54	CHN5_6ECNT_REG	RW	CHN5 and CHN6 edge counter setting register
16	58	reserved		
17	5C	CHN1_PCNT_REG	RO	CHN1 PCNT output
18	60	CHN2_PCNT_REG	RO	CHN2 PCNT output
19	64	CHN3_PCNT_REG	RO	CHN3 PCNT output
1A	68	CHN4_PCNT_REG	RO	CHN4 PCNT output
1B	6C	CHN5_PCNT_REG	RO	CHN5 PCNT output
1C	70	CHN6_PCNT_REG	RO	CHN6 PCNT output
1D	74	reserved		
1E	78	reserved		
1F	7C	FECFG_REG	RW	Front-End configuration register
8000	20000	FIFO_REG	RO	Readout of FIFO memory

Copyright, © 2002 Bustec Production Ltd.

6.2 Register Description

6.2.1 FCID_REG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	1	1	0	0	0	1	1	0	1	0	0	1	0	0	0
Content	ANA	DIG	In	Out	Model						Number of Channels					

ANA 0 : indicates FC not houses analogue channels
DIG 1 : indicates FC houses digital channels
In 1 : function card with input capabilities
Out 0 : function card without output capabilities
Model 6: CTFC model identifier
Channels 4 or 6 : number of input channels of the CTFC

FCID_REG contains identification number of function card type.
Readout should give a value of:

6344H for 4-channel FC

6346H for 6-channel FC

6.2.2 FCVER_REG

This is FC version register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Content	VER[15:0]															

6.2.3 FCCTRL_REG

Function card Control and Status Register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation					RO	RO	RO	RO	RO	RO	RO		RW	RWC	RW	RWC
Initial					0	0	0	1	0	0	0		0	0	0	0
Content	Not used				COUNTING_ END	COUNTING_state	ARMED_state	ACCESS_state	PCNTS_ERR	TICNTS_ERR	OVERWRITE_ ERR	Not used	TTLOUT_EN	SW_IGATE_START	SW_GATE	FSM reset

Copyright, © 2002 Bustec Production Ltd.

FSMreset Resets internal (in FPGA) state machines. The reset doesn't change content of registers.
Reset is started by writing "1" to that bit. After the reset is done, the hardware clears the bit. Software should poll the bit until it is cleared.
It is recommended to perform reset during FC initialisation.
Write
0: no effect
1: starts reset of internal state machine
Read
0: reset finished (if reset previously started)
1: reset in progress
USAGE
- During initialisation process as first step
- To force state machine to known (ACCESS_state) state
- To stop data acquisition before gate is ended
- FSMreset takes approximately 1µs
- FSMreset bit automatically initiates CLEARING_CMD and FIFO_reset

SW_GATE The bit is used to generate gate signal by software.
Write
0: the GATE signal is off
1: the GATE signal is on
Read
gives the last written value
USAGE
- The action on this bit has effect only if GATE_SEL is set to select software gate
- This bit is cleared when FSMreset is performed

SW_IGATE_START The bit is used to start internal gate signal by software.
Write
0: no changes
1: generates pulse to start IGATE
Read
Always gives zero
USAGE
- The action on this bit has effect only if IGATE_START_SEL was set to select internal gate started by software

 Copyright, © 2002 Bustec Production Ltd.

TTLOUT_EN The bit is used to enable the output driver on GATEIN pin on connector. The pin becomes then the output pin and internal gate signal is driven to it.

Write

0: disables the output driver on GATEIN pin

1: enables the output driver on GATEIN pin

Read

gives the last written value

USAGE

- Used only with 4-channel FC (ID = 6844H)
- To send out the internal gate signal to front panel
- When TTLOUT_EN bit is set the GATEIN pin becomes the output

OVERWRITE_ ERR The bit is read only and is set by hardware after overwrite of data (due to the bottleneck on FIFO input) from TICNT has occurred.

This bit is cleared on arming command, clearing command or by FSMreset bit.

Write

No effect

Read

0: no OVERWRITE errors

1: OVERWRITE error has occurred

USAGE

- This bit is used to detect if overwrite of data from TICNT happened

TICNTS_ERR The bit is read only and is set by hardware if a double full revolution of any TICNT without storing data in FIFO occurred.

This bit is cleared on arming command, clearing command or by FSMreset bit.

Write

No effect

Read

0: no errors from TICNT

1: error in any TICNT has occurred

USAGE

- This bit is used to detect if TICNT error occurred

PCNTS_ERR The bit is read only and is set by hardware after full revolution of PCNT has occurred.

This bit is cleared on arming command, clearing command or by FSMreset bit.

Write

No effect

Read

0: no errors from PCNT

1: error in any PCNT has occurred

USAGE

- This bit is used to detect if full revolution of PCNT has happened

Copyright, © 2002 Bustec Production Ltd.

ACCESS_state The bit indicates ACCESS_state of internal state machine.
Write
No effect
Read
0: internal SM in state other than ACCESS_state
1: internal SM in ACCESS_state
USAGE
- To detect state of internal state machine
- ACCESS_state is the initial state of internal SM (an idle state). Card configuration is allowed only in this state.

ARMED_state The bit indicates ARMED_state of internal state machine.
Write
No effect
Read
0: internal SM in state other than ARMED_state
1: internal SM in ARMED_state
USAGE
- To detect state of internal state machine
- In armed state the card waits for the gate signal

COUNTING_state The bit indicates COUNTING_state of internal state machine.
Write
No effect
Read
0: internal SM in state other than COUNTING_state
1: internal SM in COUNTING_state
USAGE
- To detect state of internal state machine
- The counting can take place in this state only. It leaves this state when gate ends or when FSM reset is performed

COUNTING_ END The bit is read only and is set by hardware after end of measurement process, when gate signal becomes inactive.
This bit is cleared on arming command or clearing command.
Write
No effect
Read
0: counting in progress (if previously started)
1: counting ended
USAGE
- This bit can be used to detect end of measurement process. There are possible two ways: polling the bit or waiting for interrupt generated by this bit when output trigger was enabled

 Copyright, © 2002 Bustec Production Ltd.

6.2.4 FIFOCTRL_REG

This register is a control/status register of the FIFO memory.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO					RWC	RWC
Initial	0	0	0	0	0	0	0	0	0	1					0	0
Content	FIFO_STATUS[7:0]								FIFO_FULL	FIFO_EMPTY	Not used				FIFO_WR	FIFO_reset

FIFOreset The bit resets the FIFO. Reset is done by writing "1" to that bit and waiting for "0". Resetting the FIFO means clearing the status of the FIFO and setting the empty flag.

Write
- 0: no effect
- 1: starts reset of FIFO

Read
- 0: resetting finished (if previously started)
- 1: resetting in progress

USAGE
- To emptying the FIFO
- The FSMreset resets FIFO as well

FIFO_WR The bit launches write of the data stored in IGATEL_REG and IGATEH_REG to the FIFO.

Write
- 0: no effect
- 1: starts write to the FIFO

Read
- 0: write to FIFO finished (if previously started)
- 1: write to FIFO in progress

USAGE
- To test the FIFO by writing data and reading it back

FIFO_EMPTY The bit indicates that FIFO memory is empty.

Write
- No effect

Read
- 0: FIFO not empty
- 1: FIFO empty

USAGE
- To detect if the FIFO is empty when moving data from FIFO

Copyright, © 2002 Bustec Production Ltd.

FIFO_FULL

The bit indicates that FIFO memory is full.

Write

No effect

Read

0: FIFO not full

1: FIFO full

USAGE

- To detect if the FIFO is full when moving data from FIFO

FIFO_STATUS[7:0]

The bits indicate the contents of the FIFO memory.

Write

No effect

Read

Gives the amount of samples stored in FIFO. The FIFO_STATUS[7:0] should be multiplied by 4 to get the approximate number of samples. The maximum inaccuracy in determining the exact number of samples in FIFO is lower than 4.

USAGE

- To detect how many samples are in FIFO
- FIFO_STATUS[7:0] *4 ≤ Number_of_samples_in_FIFO < (FIFO_STATUS[7:0]+1) *4
- When the FIFO_STATUS[7:0] is zero and EMPTY flag is not set that means that FIFO has 1, 2 or 3 samples. The software should read out one sample and check the EMPTY bit. Remember that read out of the 32-bit sample composes of two 16-bit word readouts.

6.2.5 COMMAND_REG

Write to this register performs arming command: CTFC card goes from ACCESS_state to ARMED_state. The data doesn't matter.

Read from that register performs clearing command: it clears the error bits and the output trigger line. The data doesn't matter.

6.2.6 OTRI_REG

Output Trigger register allows to select the source of the output trigger sent to the motherboard and farther to VXI controller.

There are following trigger sources:

- Errors (PCNT_ERR, TICNT_ERR, OVERWRITE_ERR)
- End of counting
- Reaching the pre-set number of samples in FIFO

The hardware allows the use of more than one trigger source at the time.

The Output Trigger can be used to generate interrupt at the end of measurement process.

Copyright, © 2002 Bustec Production Ltd.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0			0	0	0	0	0
Content	OTRIG_ status	PCNTS_ERR_EN	TICNTS_ERR_EN	OVERWRITE_ ERR_EN	ERR2OTRIG_EN	CEND2OTRIG_EN	OTRIG_EN	OTRIG_LEVEL	SW_OTRIG	Not used		FIFO2OTRIG_EN	FIFO_TH[3:0]			

FIFO_TH[3:0] The bits set the number of samples stored in FIFO needed to generate the trigger.

Write

FIFO_TH[3:0] x 64 gives the number of samples which are required to be stored in FIFO in order to generate the trigger

Read

Gives the last written value

USAGE

- In order to enable trigger generated by FIFO the FIFO2OTRIG_EN and the OTRIG_EN bit should be set

FIFO2OTRIG_EN The bit enables generating output trigger when selected on FIFO_TH[3:0] bits number of samples in FIFO was reached.

Write

0: generating trigger after reaching set number of samples disabled

1: generating trigger after reaching set number of samples enabled

Read

Gives the last written value

USAGE

- In addition to this bit the OTRIG_EN bit should be set to enable output trigger

SW_OTRIG The bit allows setting output trigger by software.

Write

0: no change

1: sets the output trigger

Read

Gives the status of software generated output trigger

USAGE

- This trigger source was designed for debugging purposes only

Copyright, © 2002 Bustec Production Ltd.

OTRIG_LEVEL The bit allows to select the way of output trigger generating mode:

- Pulse – after rising edge of trigger source pulse of 400-800ns width will be generated independently of trigger source high level duration
- Level – after rising edge of trigger source output trigger level will follow the level of trigger source.

Write

0: output trigger generating mode set to pulse (400-800ns width)

1: output trigger generating mode set to level

Read

Gives the last written value

USAGE

- When working with interrupts the level mode should be set

OTRIG_EN The bit is the main output trigger enable bit. If this bit is cleared no output trigger will be sent to MB independently of trigger source enable bits. If this bit is set output trigger will be sent if any of trigger source is enabled and its condition is met.

Write

0: output trigger disabled

1: output trigger enabled

Read

Gives the last written value

USAGE

- Main output trigger enabling bit

CEND2OTRIG_EN The bit enables generating output trigger when measurement process was ended (gate signal became inactive).

Write

0: generating trigger on counting end disabled

1: generating trigger on counting end enabled

Read

Gives the last written value

USAGE

- In addition to this bit the OTRIG_EN bit should be set to enable output trigger

ERR2OTRIG_EN The bit enables generating output trigger when error occurred.

Write

0: generating trigger on error condition disabled

1: generating trigger on error condition enabled

Read

Gives the last written value

USAGE

- In addition to this bit the OTRIG_EN bit should be set to enable output trigger and particular error source should be enabled

 Copyright, © 2002 Bustec Production Ltd.

OVERWRITE_ ERR_EN

The bit enables OVERWRITE_ERR as an error condition.

Write

0: OVERWRITE_ERR disabled

1: OVERWRITE_ERR enabled

Read

Gives the last written value

USAGE

- This bit is used to enable measurement to be stopped on overwrite error (if ERR_STOPED_EN was set) or to sent the trigger to MB when the error happened (when ERR2OTRIG_EN and OTRIG_EN bits were set)

TICNTS_ERR_EN

The bit enables TICNTS_ERR as an error condition.

Write

0: TICNTS _ERR disabled

1: TICNTS _ERR enabled

Read

Gives the last written value

USAGE

- This bit is used to enable measurement to be stopped on TICNT error (if ERR_STOPED_EN was set) or to sent the trigger to MB when the error happened (when ERR2OTRIG_EN and OTRIG_EN bits were set)

PCNTS_ERR_EN

The bit enables PCNTS_ERR as an error condition.

Write

0: PCNTS_ERR disabled

1: PCNTS_ERR enabled

Read

Gives the last written value

USAGE

- This bit is used to enable measurement to be stopped on PCNT error (if ERR_STOPED_EN was set) or to sent the trigger to MB when the error happened (when ERR2OTRIG_EN and OTRIG_EN bits were set)

OTRIG_status

The state of the output trigger line.

Write

No effect

Read

0: output trigger inactive

1: output trigger active

USAGE

- When working with interrupts this bit should be used by the interrupt routing to determine the interrupt source

Copyright, © 2002 Bustec Production Ltd.

6.2.7 ITRI_REG

The register allows selection of the common trigger source used optionally to start TI counters.
There are possible following input trigger sources:

- Software trigger
- MB input trigger
- External trigger through FP

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation									RO				RW	RW	RW	RW
Initial									X				X	X	X	X
Content	Not used								COMTRIG_status	Not used			SW_COMTRIG	COMTRIG_SEL[1:0]		FP_COMTRIG_ALOW

FP_COMTRIG_ALOW The bit enables changing of the active level of external common trigger coming from FP.
Write
0: active level high
1: active level low
Read
Gives the last written value

COMTRIG_SEL[1:0] The bits select the source of common trigger.
Write
00: software generated common trigger
01: common trigger from MB
10: common trigger from external connector
11: reserved
Read
Gives the last written value

COMTRIG_status The state of common trigger.
Write
Has no effect
Read
Gives the current state of the common trigger

Copyright, © 2002 Bustec Production Ltd.

6.2.8 DAC_REG

The DAC_REG allows setting of output value of the on-board DAC. The 8-output DAC is used to control offset of each channel.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RWC		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Initial	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	DAC trans	Not used	DAC_ADDR[3:0]				DAC_DATA[9..0]									

DAC_DATA[9..0] The bits are data to write to 8-channel DAC. These bits set DAC outputs.

Write

DD[9..0] set the output value of DAC

Read

These are write only bits

USAGE

- The data should be in binary format
- The DAC outputs range is –5.00V..+4.99V
- Output voltage = 5*(DD[9..0] – 512)/512 and is expressed in volts

DAC_ADDR[3..0] The bits address the channel of DAC.

Write

DAC_ADDR[3..0] set the address of DAC channel

Read

These are write only bits

USAGE

- DAC_ADDR[3..0] = Channel_number.
- Channel_number range is from 1 to 6

DACtrans The bit starts data shifting out to DAC. Writing “1” to this bit starts data (D[12..0]) shifting out to DAC. This bit is cleared to “0” after shifting is finished.

Write

0: no effect

1: starts DAC_DATA[9..0] shifting out

Read

0 : shifting out to DAC finished (if previously started)

1 : shifting out to DAC in progress

USAGE

- To start and detect the end of shifting out
- Shifting out takes approximately 8µs
- During shifting out DACtrans bit is set

Copyright, © 2002 Bustec Production Ltd.

6.2.9 MODE_REG

This is mode register. Each bit of this register is write and read able.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	Reserved						PCNT_UPWORD	ERR_STOPED_EN	TB_SEL[2:0]			TB_EN	IGATE_START_SEL	GATE_SEL[1:0]		GATEIN_ALOW

GATEIN_ALOW The bit sets the active level of external gate.
Write
0: active level high
1: active level low
Read
Gives the last written value

GATE_SEL[1:0] The bits select the source of gate signal.
Write
00: software generated gate
01: external gate
10: internal gate
11: gate disabled
Read
Gives the last written value

IGATE_START_SEL The bit selects the source that starts internal gate generation.
Write
0: internal gate started by software
1: internal gate started from external signal applied to external gate pin
Read
Gives the last written value

TB_EN The bit enables time base for Time Interval counters.
Write
0: Time Base disabled
1: Time Base enabled
Read
Gives the last written value

Copyright, © 2002 Bustec Production Ltd.

TB_SEL[2:0] The bits set the frequency of the time base of TI counters.
Write
000: time base set to 80MHz
001: time base set to 8MHz
010: time base set to 800KHz
011: time base set to 80KHz
100: time base set to 8KHz
101: time base set to 800Hz
110: reserved
111: reserved
Read
Gives the last written value

ERR_STOPPED_EN The bit enables stopping the measurement when an error happens.
Write
0: errors don't stop measurement
1: errors stop measurement
Read
Gives the last written value
USAGE
- In addition to this bit error source has to be selected and enabled (bits OVERWRITE_ERR_EN, TICNTS_ERR_EN and PCNTS_ERR_EN in OTRI_REG)

PCNT_UPWORD The bit allows the switching between upper and lower word of the PCNT output.
Write
0: lower word selected to read out through the CHNx_PCNT_REG
1: upper word selected to read out through the CHNx_PCNT_REG
Read
Gives the last written value
USAGE
- The content of PCNT is read out in two steps: lower word and upper word. The PCNT_UPWORD bit selects the word which can be readout from the CHNx_PCNT_REG

6.2.10 IGATEx_REG

The registers allow to set-up the internal gate width. The bits IGD[31..0] of 32-bit internal gate counter are set through IGATEL_REG and IGATEH_REG. For testing purposes the contents of these registers is loaded to FIFO when FIFO_WR was set.

IGATEL_REG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	IGD[15..0]															

IGATEH_REG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	IGD[31..16]															

IGD[31..0] The bits set the width of internal gate.
Write
Set the internal gate width
Read
Gives the last written value

The internal gate width is expressed by the following equation:

$$T_{IGATE} = T_{OSC} \cdot (IGD[31..0])$$

where

T_{IGATE} – the internal gate width
T_{OSC} – it is period of signal clocking the internal gate counter, equal 400ns
IGD[31..0] – it is value written to IGATEL_REG and IGATEH_REG

To calculate value, which has to be written (after rounding to integer) to register the following equation can be used:

$$IGD[31..0] = \frac{T_{IGATE}}{T_{OSC}}$$

(description as above).
The range of IGD[31..0] is from 1 to FFFF_FFFFH giving the value of T_{IGATE} from 400ns to 1717.98s.

6.2.11 CHNx_CFG_REG

This is channel configuration register. The “x” represents channel number and changes from 1 to 6.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0			0	0	0	0	0	0
Content	CHNx_PCNT_ERR	LIMITED_COMPLETED	CHNx_TRIG_SEL	CHNx_TRIGIN_ALOW	CHNx_SYNC	CHNx_LIMITED	CHNx_WINDOW	CHNx_TRIG_STARTED	Not used		CHNx_REDGE_FIRST	CHNx_FEDGE_EN	CHNx_REGDE_EN	CHNx_PCNT_FEDGE	CHNx_PCNT_EN	CHNx_EN

CHNx_EN The bit enables channel #x input.
Write
0: channel disabled
1: channel enabled

Copyright, © 2002 Bustec Production Ltd.

Read
Gives the last written value

CHNx_PCNT_EN The bit enables the PCNT in channel #x.
Write
0: PCNT disabled
1: PCNT enabled
Read
Gives the last written value

CHNx_PCNT_FEDGE The bit sets the edge direction the PCNT in channel #x reacts on.
Write
0: PCNT counts rising edges
1: PCNT counts falling edges
Read
Gives the last written value

CHNx_REDGE_EN The bit enables raising edges as an events for TICNT in channel #x.
Write
0: raising edges disabled
1: raising edges enabled
Read
Gives the last written value

CHNx_FEDGE_EN The bit enables falling edges as an events for TICNT in channel #x.
Write
0: falling edges disabled
1: falling edges enabled
Read
Gives the last written value

CHNx_REDGE_FIRST The bit defines the starting edge for TICNT in channel #x.
Write
0: TICNT starts counting from falling edge
1: TICNT starts counting from raising edge
Read
Gives the last written value
USAGE
- This bit has to be set when both rising and falling edges are enabled as an event (CHNx_REDGE_EN and CHNx_FEDGE_EN bits set)

CHNx_TRIG_STARTED When the bit is set counting in TICNT in channel #x is started by trigger signal. (while gate is active)
Write
0: TICNT starts immediately after gate
1: TICNT starts when trigger is active during gate active
Read
Gives the last written value

CHNx_WINDOW When the bit set the mode of trigger for channel #x: launch or window.
Write
0: launch mode is set (trigger starts only the counting)
1: window mode is set (counting is enabled during trigger)

Copyright, © 2002 Bustec Production Ltd.

Read
Gives the last written value

CHNx_LIMITED The bit sets the limited mode of TICNT for channel #x.
Write
0: limited mode is disabled
1: limited mode is enabled
Read
Gives the last written value

CHNx_SYNC The bit sets the synchronous mode of TICNT for channel #x.
Write
0: synchronous mode is disabled
1: synchronous mode is enabled
Read
Gives the last written value

CHNx_TRIGIN_ALOW The bit enables setting of the active level of external independent trigger coming from FP.
Write
0: active level high
1: active level low
Read
Gives the last written value

CHNx_TRIG_SEL The bit selects the trigger for channel #x: common trigger or independent trigger.
Write
0: common trigger selected
1: independent trigger selected
Read
Gives the last written value

LIMITED_COMPLETED The bit indicates that set number of samples has been acquired for the channel #x when working in limited mode.
Write
No effect
Read
0: set number of samples not reached
1: set number of samples reached
USAGE
- This bit is cleared on arming command

CHNx_PCNT_ERR The bit gives the overflow error from PCNT for the channel #x.
Write
No effect
Read
0: no overflow error for PCNT
1: overflow happened and data from PCNT for the channel #x are not valid
USAGE
- This bit is cleared on arming command or clearing command

 Copyright, © 2002 Bustec Production Ltd.

6.2.12 CHN1_2ECNT_REG

The register allows setting of edge counter in channel 1 and channel 2. The edge counter defines the number of edges (events) to be collected in LIMITED mode.
When set in Limited mode each channel can store up to 256 values in FIFO memory, depending on setting on CHNxEC[7:0] bits. The number of data written to FIFO for all channels can not exceed FIFO size minus 1 (1023 samples) if FIFO is not emptying during measurement process.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	CHN2EC[7:0]								CHN1EC[7:0]							

CHN2EC[7:0] The bits set the number of input signal edges that has to be acquired as a latch event in TICNT.
Write
Sets the ECNT in channel #1
Read
Gives the last written value
USAGE
This bits have to be set in Limited mode only

CHN1EC[7:0] The bits set the number of input signal edges that has to be acquired as a latch event in TICNT.
Write
Sets the ECNT in channel #2
Read
Gives the last written value
USAGE
- This bits have to be set in Limited mode only

The number of edges that will be counted is expressed by the following equation:

$$Number\_of\_edges = CHNxEC[7:0] + 1$$

where

x – the channel number

CHNxEC[7..0] – it is value written to ECNT in channel #x

6.2.13 CHN3_4ECNT_REG

The register allows setting of edge counter input data in channel 3 and channel 4. The edge counter defines the number of edges (events) to be collected in LIMITED mode.
When set in Limited mode each channel can store up to 256 values in FIFO memory, depending on setting on CHNxEC[7:0] bits. The number of data written to FIFO for all channels can not exceed FIFO size minus 1 (1023 samples) if FIFO is not emptying during measurement process.

Copyright, © 2002 Bustec Production Ltd.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	CHN4EC[7:0]								CHN3EC[7:0]							

CHN3EC[7:0] The bits set the number of input signal edges that has to be acquired as a latch event in TICNT.
Write
Sets the ECNT in channel #3
Read
Gives the last written value
USAGE
This bits have to be set in Limited mode only

CHN4EC[7:0] The bits set the number of input signal edges that has to be acquired as a latch event in TICNT.
Write
Sets the ECNT in channel #4
Read
Gives the last written value
USAGE
- This bits have to be set in Limited mode only

The number of edges that will be counted is expressed by the following equation:

$$Number\_of\_edges = CHNxEC[7:0] + 1$$

where
x – the channel number
CHNxEC[7..0] – it is value written to ECNT in channel #x

6.2.14 CHN5_6ECNT_REG

The register allows setting of edge counter input data in channel 5 and channel 6. The edge counter defines the number of edges (events) to be collected in LIMITED mode.
When set in Limited mode each channel can store up to 256 values in FIFO memory, depending on setting on CHNxEC[7:0] bits. The number of data written to FIFO for all channels can not exceed FIFO size minus 1 (1023 samples) if FIFO is not emptying during measurement process.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	CHN6EC[7:0]								CHN5EC[7:0]							

CHN5EC[7:0] The bits set the number of input signal edges that has to be acquired as a latch event in TICNT.
Write
Sets the ECNT in channel #5
Read
Gives the last written value
USAGE
This bits have to be set in Limited mode only

Copyright, © 2002 Bustec Production Ltd.

CHN6EC[7:0] The bits set the number of input signal edges that has to be acquired as a latch event in TICNT.
Write
Sets the ECNT in channel #6
Read
Gives the last written value
USAGE
- This bits have to be set in Limited mode only

The number of edges that will be counted is expressed by the following equation:

$$Number\_of\_edges = CHNxEC[7:0] + 1$$

where
x – the channel number
CHNxEC[7..0] – it is value written to ECNT in channel #x

6.2.15 CHNx_PCNT_REG

The registers are read only and allow to readout the value of Pulse Counter in channel x. The 32-bit value from Pulse Counter is readout as two 16-bit numbers from CHNx_PCNT_REG and should be then merged. The selection between upper and lower word that is seen in CHNx_PCNT_REG is done using PCNT_UPWORD bit from MODE_REG.
The "x" value represents channel number and is in range from 1 to 6.

CHNx_PCNT_REG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	CHNxPC[31..16] or CHNxPC[15..0] depending on PCNT_UPWORD bit setting															

CHNxPC[31..0] The value from PCNT counter in channel #x.
x: 1 to 6
Write
Has no effect
Read
Gives the current value from Pulse Counter in channel #
USAGE
- To readout the value stored in the Pulse Counter
- Readout of this register during the measurement can give incorrect value (due to changes of counter bits)

Copyright, © 2002 Bustec Production Ltd.

6.2.16 FECONF_REG

This register allows configuring the front-end of the CTFC.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Content	Reserved	Reserved	Reserved	Reserved	CHN6_TERM	CHN6_DC	CHN5_TERM	CHN5_DC	CHN4_TERM	CHN4_DC	CHN3_TERM	CHN3_DC	CHN2_TERM	CHN2_DC	CHN1_TERM	CHN1_DC

CHx_DC
x: 1 to 6

The bit allows the input of channel #x to be DC coupled.

Write

0: relay with DC coupling path closed – DC coupling on

1: relay with DC coupling path opened – AC coupling on

Read

Gives the last written value

USAGE

•

CHx_TERM
x: 1 to 6

The bit allows the input of channel #x to be terminated with 50Ω.

Write

0: 50Ω terminator connected - 50Ω termination

1: 50Ω terminator not connected – 1MΩ termination

Read

Gives the last written value

6.2.17 FIFO_REG

The FIFO memory contains the data from TICNT counters. The 32-bit word is written into FIFO. The format of this data is as follow:

FIFO_DATA[31:29]	Channel number (0 means channel number 1)
FIFO_DATA[28:27]	Not used
FIFO_DATA[26]	OVER_ERR: When 1 overwrite error occurred for this sample
FIFO_DATA[25]	TICNT_ERR: When 1 TI counter error occurred for this sample
FIFO_DATA[24]	FR: Full revolution bit
FIFO_DATA[23:0]	TICNT_DATA: Data from TICNT counters

Every sample stored in FIFO has to be read in two steps: lower 16-bit word as a first readout and upper 16-bit word as a second readout. For the given number of samples written to the memory double number of 16-bit wide readouts has to be performed.
After moving the samples from the FIFO to host the samples have to be rearranged according to the channel number. Then to get the time interval value the following equation has to be used:

 Copyright, © 2002 Bustec Production Ltd.

$$T_n = \left(TICNT\_DATA_n - TICNT\_DATA_{n-1} + 16777216 * \left|FR_n - FR_{n-1}\right|\right) * \left(1 - TICNT\_ERR_n\right) * (1 - OVER\_ERR_n)$$

where

n – sample index, for n=0 (first data) TICNT_DATA $_{-1}$ should be set to zero,

| | - means absolute value.

T_n takes value of zero if TICNT_ERR$_n$ or OVER_ERR$_n$ happened (the bits are set to one). In such case this value has to be rejected.

Copyright, © 2002 Bustec Production Ltd.

 Copyright, © 2002 Bustec Production Ltd.

Bustec Production, Ltd.
World Aviation Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

Bustec, Inc.
17820 Englewood Dr #14, Middleburg Hts, OH 44130, U.S.A
Tel. +1 440 826 4156, Fax: +1 440 826 4184